



## Optimal multi-block mesh generation for CFD

Journal:	<i>International Journal of Computational Fluid Dynamics</i>
Manuscript ID	GCFD-2017-0021.R3
Manuscript Type:	Full Paper
Keywords:	Multi-Block Structured Mesh Generation, Medial Axis Transform, Automatic Blocking, Distance Field/level sets, Adjoint Based Error Analysis

SCHOLARONE™  
Manuscripts

To appear in the *International Journal of Computational Fluid Dynamics*  
Vol. 00, No. 00, 00 Month 20XX, 1–25

## *Optimal multi-block mesh generation for CFD*

(Received 00 Month 20XX; final version received 00 Month 20XX)

An assessment of various automatic block topology generation techniques for creating structured meshes has been performed in the first part of the paper. The objective is to find out optimal blocking methods for generating meshes suitable for flow simulations. The comparison has been carried out using an adjoint based error analysis of the meshes generated by these block topologies. Different objective functions and numerical schemes have been used for this assessment. It is found that, in general, the medial axis based approaches provide optimal blocking and yields better accuracy in computing the functional of interest. This is because the medial axis based methods produce meshes which have better flow alignment specially in case of internal flows. In the second part of the paper, the adjoint based error indicator has been used to adapt the block topology in the regions of large error.

**Keywords:** Multi-Block Structured Mesh Generation; Medial Axis Transform; Automatic Blocking; Distance Field/level sets; Adjoint Based Error Analysis.

### 1. Introduction

#### 1.1. *Structured Quad/Hex Meshing*

Computational fluid dynamics (CFD) has now become an essential part of the modern day design hierarchy. Each step in the CFD design process from the CAD definition to the optimum design needs to be as computationally efficient and reliable as possible. Mesh generation has remained one of the most time consuming steps in the CFD process and faces even bigger challenges with the ever increasing need for simulating complex three dimensional (3D) flows. There is a trade-off between the mesh quality, ease of generation, solver requirements and parallel mesh generation when choosing amongst the structured and unstructured mesh types. Unstructured meshes offer more flexibility for meshing complex 3D domains. Structured meshes, on the other hand, offer higher numerical accuracy and less storage than the unstructured meshes. They also allow relatively easy implementation of high-order finite volume and finite difference schemes.

#### 1.2. *Multi-block Topology Generation*

Multi-block structured mesh generation is among the most widely used meshing techniques in flow simulations. Multi-blocking is essentially a two-stage process. In the first stage, a suitable blocking topology is generated which divides the complex domain into simple sub-domains. The resulting blocks are subsequently meshed. This structured blocking offers an efficient meshing strategy for topologically simple configurations like the main gas flow path and standard templates exist for partitioning of such domains. For example, the H-O-H type blocking is commonly used to mesh the blade passage as shown in the center of the Figure 1(a) (Shahpar and Lapworth 2003; Milli and Shahpar 2012). However, the modern day design challenges demand the computational analysis of more realistic geometries. A turbine blade, for example, has cooling holes, internal cooling passages, cut back trailing edges, shroud cavities and rim seals as depicted in the Figure 1(a). Examples of other complex flow domains include, for instance, a racing car geometry for aerodynamics analysis, an electronics system for thermal and ventilation analysis and a wind farm domain for an efficient

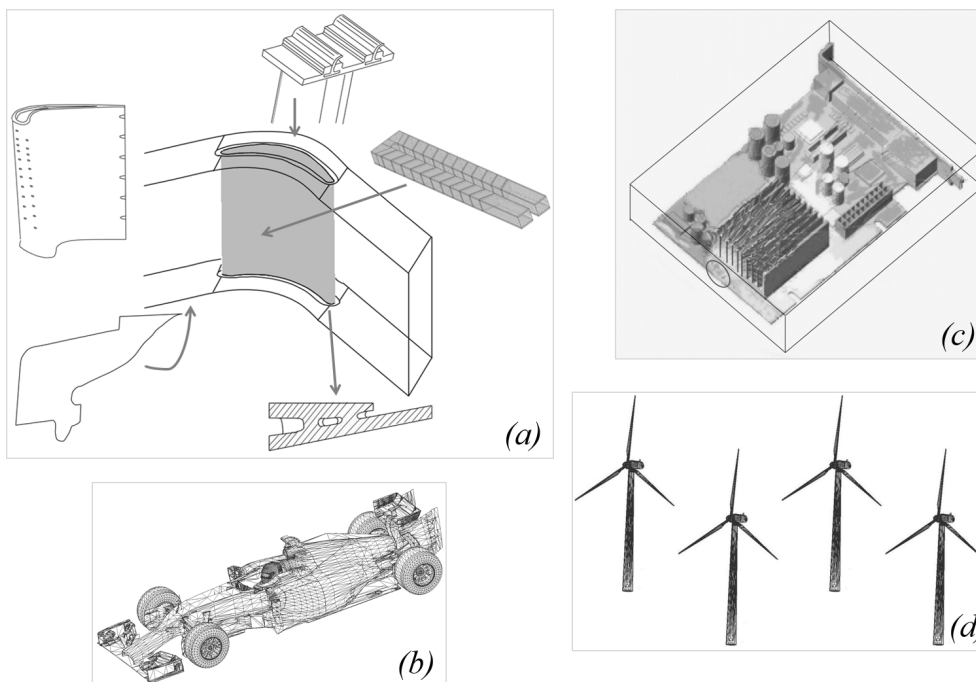


Figure 1. Meshing challenges: (a) Turbine blade (counter-clockwise from top left) cooling holes, seal cavity, cutback trailing edges, internal cooling passages, shroud cavity, centre: standard H-O-H blocking for the blade passage (Shahpar and Lapworth 2003; Milli and Shahpar 2012). (b) a racing car geometry for aerodynamics analysis (c) electronics systems for thermal and ventilation analysis (d) a wind farm domain for efficient wind turbine placement study.

wind turbine placement study, as shown in frames (b), (c) and (d) of the Figure 1 respectively. Meshing such multiply linked and more diverse geometries requires significant user intervention, or writing of templates as part of a library (Milli and Shahpar 2012). Thus, an automatic blocking strategy can be beneficial to reduce the CFD design cycle time and could be a better alternative to the unstructured or hybrid meshing methods.

Fully automatic 3D block topology generation is a complex problem and currently there is no ideal block topology algorithm with the desired features for structured mesh generation. However various automatic blocking approaches have been proposed with varying levels of automation and geometric complexity handling. This include approaches based on medial axis (Tam and Armstrong 1991; Price and Armstrong 1995a,b; Sheehy, Armstrong, and Robinson 1995), paving/plastering (Blacker and Myers 1993; Owen and Blacker 2005) and more recently methods based on cross/frame field (Fogg, Armstrong, and Robinson 2015a; Kowalski, Ledoux, and Frey 2014).

The medial axis transform (MAT) based algorithms for the domain decomposition have been presented in, for example, by Armstrong and collaborators (Tam and Armstrong 1991; Price and Armstrong 1995a,b; Sheehy, Armstrong, and Robinson 1995). Here the medial axis is generated using the Voronoi based method. The constrained Delaunay triangulation of points distributed on the boundary is then used to assemble shape molecules featuring a mesh topology. The shape molecules are reduced to shape atoms by inserting cuts between the medial vertices which topologically represent 4-, 5- and 6-sided polygons. A subdivision is created resulting in one block for each medial vertex, medial edge and medial face. A midpoint subdivision is then used for meshing the blocks. The concave features are removed by splitting the edges which are chosen based on a corner angle criteria. An alternative has been presented by Rigby (Rigby 2004), called the TopMaker approach, which makes use of medial vertices and parts of medial axis to block the domain. Medial vertices are defined as the points which are equidistant from three locations form the domain boundary. Consequently, six types of medial edges and appropriate rules are defined for creating the blocks. Further enhancements have been included to produce a good quality mesh however this technique has yet to be extended for 3D.

Distance field based approaches are also widely used for the medial axis approximation. The 2D Euclidean distance map based approach involves an analysis of the local directional maxima to obtain the medial axis (Danielsson 1980; Ragnemalm 1993). A nearest-neighbor search based criteria to perform a spatial subdivision for approximating the medial axis is used by Vleugels and Overmars (Vleugels and Overmars 1995). Another hybrid approach called differential MAT or d-MAT approach is presented by Xia and Tucker (Xia and Tucker 2010). The hyperbolic-natured eikonal, level set equation is used to calculate the distance field (Tucker 2003). Medial axis point clouds are then extracted from the Laplacian or Hessian determinant of the distance field. A thinning algorithm is used for thinning the point clouds into curves and surfaces. Such a hybrid approach thus avoids complexity of pure geometric approach and provides more accuracy than the pure image thinning algorithms. A further enhancement to this approach is a biased MAT proposed by Xia and Tucker (Xia and Tucker 2011). This adds more flexibility to the subsequent domain decomposition.

LayTracks3D (Quadros 2014), is a hybrid hexahedral meshing method combining medial axis based decomposition and the advancing front method. Here, the medial axis is used directly to create the mesh instead of the domain decomposition. Regularly spaced thin strips are created between the medial radii pairs from the medial axis to the associated touch points. A further subdivision is then carried out to form 'Tracks'. The advancing front method is then applied from the boundary towards the medial axis to generate the mesh. This technique produces good quality hexahedral meshes but degenerate cells can be formed around the sharp concave features.

Recent advancements in mesh generation are the methods based on the cross-fields (frame fields in 3D). A cross field is defined by assigning a set of four unit vectors to points at the discrete locations. These unit vectors form a regular cross on the tangent plane. Thus the size and the orientation of the quadrilateral cells can be specified by the cross field. A number of approaches have been put forward for 2D and 3D cross field based domain decomposition and mesh generation. To generate the block topology, the partitioning created by connecting the cross-field streamlines to the singularities can be used. The resulting blocks of the cross field can then be mapped to a grid (Bommes, Lempfer, and Kobbelt 2011; Li et al. 2011; Palacios and Zhang 2007; Kowalski, Ledoux, and Frey 2013).

Fogg et al. (Fogg, Armstrong, and Robinson 2015a) also use the cross field for block topology generation. Their method is similar to the paving algorithm. The cross field is initialized at the boundaries and then propagated inwards using the fast marching method. A continuous cross field is obtained up to the location where the advancing fronts collide giving rise to singularities and the resulting multi-block decomposition. Such a method is simpler than the paving algorithm because the mesh connectivity is not catered in the cross field. A variation of this method involves a medial axis based blocking technique with implicit use of the cross field (Fogg, Armstrong, and Robinson 2015b). This results in effective handling of the concavities and sharp features as compared to the pure medial axis based techniques such as TopMaker.

Kowalski et al. (Kowalski, Ledoux, and Frey 2013) use a PDE based approach to obtain the cross field. A diffusion problem is solved for this purpose. After locating the field singularities, the lines in the cross field connecting the singularities are extracted that result in the domain partitioning. Huang et al. (Huang et al. 2011) and Li et al. (Li et al. 2012) use energy function minimization to smooth the frame fields. However, here a rough initialization of the surface cross field is used which can result in poor singularity locations and hence can deter the generation of all hexahedral meshes. Kowalski et al. (Kowalski, Ledoux, and Frey 2014) describe a frame field method that does not require initial surface cross field. The cross field approach towards domain decomposition and mesh generation is novel and efficient but quite complex and expensive.

Malcevic (Malcevic 2011) presents another automated blocking strategy based on a Cartesian fitting method. While preserving the topology definition, a forward geometry simplification is performed followed by fitting the model into a Cartesian framework. The next step is blocking the domain after which the blocked model is mapped back on to the original geometry. Further operations such as removing singularities by J-grid wrapping are performed to enhance the mesh

quality. This technique has been applied for meshing the end-wall cavities found in turbomachinery. This technique is very simple and but has only been demonstrated for 2D cases so far. The method sometimes produces some unnecessary mesh clustering across the block interfaces.

### 1.3. Error Estimation

Several approaches have been previously presented in the context of error estimation. A review of these approaches is presented by Roy (Roy 2010). The error estimators can be divided into two main categories: solution based and adjoint based. The solution based error indicators are based on the flow gradients or undivided differences to adapt the certain flow features like shock waves, wakes, boundary layers and slip lines etc (Baker 1997; Baskett and Haines 2001; Pirzadeh 1999; White and Corfield 2006). However, adaptation based on such indicators might not guarantee a reduction in the overall error estimate and can lead to erroneous results (Roy 2009; Dwigh 2008; Venditti and Darmofal 2002a; White and Corfield 2006).

The solution based error estimators are based on local measures of error. However, the other parts of the domain might have a strong influence on that local error estimate. Hence as stated above, the global error might not reduce by treating the local error. This is the case, for example, in convection dominated flows, where the errors upstream may affect the solution downstream. This is described in (Warren et al. 1991; Dwigh 2008), where an accurate capturing of the shock position through the local pressure gradient based error indicator was desired. The continuous local refinement based on this error estimate resulted in a shock position different to the one obtained by the uniform global refinement. This is because the error upstream of the shocks are convected and affect the strength and position of the shock. Thus engineering quantities of interest such as lift and drag become highly sensitive to the discretization and to the residual errors propagated from elsewhere in the flow field. In contrast the adjoint based error estimators can efficiently quantify the impact of such errors and cater for the propagation effects. The error estimate thus obtained can be used for accurate prediction of quantities of interest and subsequent mesh adaptation.

One of the earliest demonstrations of the adjoint error estimation and mesh adaptation is presented by Müller and Giles (Muller and Giles 2001) in which the dominant part of error correction term is expressed as the global sum of the dot product of adjoint solution and the residual error. Cells in the mesh are then refined when this dot product is greater than a certain user defined threshold. Building upon the technique described in Giles and Pierce (Giles and Pierce 2002a) and Müller and Giles (Muller and Giles 2001), Venditti and Darmofal (Venditti and Darmofal 2002a) use the additional concept of a truth mesh to estimate the error in the objective function. A truth mesh is simply a successive uniform refinement of the working coarse mesh. Error in the coarse-mesh functional is estimated with respect to its value on the fine mesh by prolongation of the coarse-mesh non-linear (also called primal) and adjoint solution on the fine grid. This strategy provides improved error estimates than the one presented by Müller and Giles (Muller and Giles 2001). This approach has been extended to 3D by Park (Park 2004). One of the difficulties associated with the approach presented by Venditti and Darmofal is the storage of the fine mesh which is an additional memory overhead. An alternate approach has been presented in Dwight (Dwight 2007) which still employs the adjoint method but the sensitivity of the objective function with respect to the added stabilizing dissipation in the numerical method is used as an error indicator. While this alleviates the need for storage of the fine mesh, the disadvantage of this strategy is that sources of error other than the added numerical dissipation are not included in the error indicator. The approaches presented in Fidkowski (Fidkowski 2007) and Nemec and Aftosmis (Nemec and Aftosmis 2007) demonstrate the use of Venditti technique for Cartesian meshes. A detailed review on the topic of error estimation using adjoint methods is presented by Fidkowski and Darmofal (Fidkowski and Darmofal 2011). Despite their advantages over the solution based error estimators, the adjoint based error estimators are complex and expensive to compute. Also, the primal solution needs to be converged in order to compute the adjoint sensitivities.

The current work compares the d-MAT (Xia and Tucker 2010), TopMaker (Rigby 2004) and the

Cartesian fitting (Malcevic 2011) based block topologies by using the adjoint based error analysis applied to 3D geometries. This is an extension of the work [name deleted to maintain the integrity of the review process] comprising mainly 2D cases. The use of this adjoint application is intended to inform template and algorithm design for multi-block meshes. Hence, the prohibitive cost and complexity of the adjoint method is not an issue. In the second part of the paper, an adjoint error based block topology adaptation method is introduced. The technique is demonstrated by using two seal geometries.

## 2. Adjoint error estimation

### 2.1. Discrete Adjoint Analysis

After the primal flow solution is available, the discrete adjoint equations are solved to get the adjoint variables. This section presents the discrete adjoint analysis. A detailed derivation can be found in Giles and Pierce (Giles and Pierce 2002b). Let  $Q$ , the flow variables at discrete set of points with coordinates  $X$ , be the solution of system of steady non-linear equations

$$R(Q, X, \alpha) = 0 \quad (1)$$

where  $R$  is the discrete residual vector and  $\alpha$  is a set of design variables. Also consider an objective function  $J(Q, \alpha)$  which one wishes to optimize. The sensitivity of this objective function to a set of design variables can be expressed in the following form

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial Q} \frac{\partial Q}{\partial \alpha} + \frac{\partial J}{\partial \alpha} \quad (2)$$

The adjoint variables  $\nu$  can be defined as the effect of the flow residual on the objective function:

$$\nu = \frac{\partial J}{\partial R} \quad (3)$$

Using Equations (1) and (3), Eqn. (2) can be written as

$$\frac{dJ}{d\alpha} = -\nu^T \frac{\partial R}{\partial \alpha} + \frac{\partial J}{\partial \alpha} \quad (4)$$

Eqn. (4) can be re-written to give the following equation for adjoint variables

$$\left( \frac{\partial R}{\partial Q} \right)^T \nu = \left( \frac{\partial J}{\partial Q} \right)^T \quad (5)$$

This set of linear equation can be solved to give the adjoint flow variables in a manner similar to the primary flow solution.

### 2.2. Adjoint Error Analysis

An adjoint error estimation procedure similar to that of Venditti and Darmofal (Venditti and Darmofal 2002b) is followed in this work. A detailed description of this method is presented in the Appendix. The global error in the objective function  $J(Q)$  can be related to the local residual error with the adjoint variables working as the weighting function (see Appendix). This relation is described by the equation

$$J(Q) - J_h(Q_h^H) = -(\nu_h^H)^T R_h(Q_h^H) \quad (6)$$

where  $H$  and  $h$  represent the coarse and the fine mesh levels respectively.

### 2.3. Flow Solvers

Simulations in this work are performed using the Rolls-Royce in-house HYDRA program which is a coupled suite of non-linear, linear and adjoint CFD solvers (Lapworth 2004). The core non-linear solver has an edge based data structure and the flow equations are integrated around median dual control volumes using a MUSCL based flux differencing scheme (Moinier 1999). It uses a block Jacobi pre-conditioner and iteration towards the steady state is carried out using a 5-stage Runge-Kutta scheme. For accelerated convergence, an element-collapsing multigrid algorithm is used.

In the original flow solver, the inviscid fluxes are computed using the flux difference scheme (FDS) of Roe with an additional smoothing term (Moinier 1999). The AUSM (advection upstream splitting method) flux vector splitting (FVS) scheme for inviscid fluxes (Liou and Steffen 1993) has been added to the solver for the assessment performed in this paper. While the FDS scheme includes an explicit smoothing term, the AUSM scheme has an implicit, advection Mach number weighted smoothing term. Hence, the amount of added numerical dissipation varies between the two schemes.

HYDRA Adjoint, the steady adjoint solver, is based upon the discrete adjoint approach (Giles et al. 2003). The gradients are evaluated using the automatic differentiation. First, the non-linear primal flow version is linearized and then the adjoint version is obtained by transposing the linearized equations. This methodology has successfully been used in design of turbomachinery components (Duta, Shahpar, and Giles 2007).

### 3. Mesh/block topology adaptation procedure

In this section the mesh movement scheme to adapt the block boundaries based on the adjoint error is described. The method involves constructing an anisotropic mesh metric based on the Hessian of the adjoint error and then moving the mesh nodes using the spring analogy. The solution variable (for example Mach number or pressure) based mesh movement scheme is already incorporated in the Rolls-Royce mesh adaptation tool. This implementation follows the work described in Ait-Ali-Yahia et al. (Ait-Ali-Yahia et al. 1996). The method is modified by replacing the scalar solution variable with the adjoint error estimate. The short description of the method is described next. The detailed procedure can be found in the original sources (Ait-Ali-Yahia et al. 1996; Habashi et al. 2000).

For a 1D problem, let  $u$  be the scalar solution (adjoint error in this case) which is approximated by  $u_h$  through linear interpolation. Then the local approximation error  $e$  over this edge of length  $h$  is given by

$$e = u - u_h \quad (7)$$

After expanding the solution  $u$  at one end of the edge and assuming zero error at the nodes, it can be shown that the interpolation error across the edge is proportional to the product of the square of edge length  $h$  and the second derivative of  $u_h$  (Ait-Ali-Yahia et al. 1996) i.e.

$$e \propto h^2 \left| \frac{d^2 u_h}{dx^2} \right| \quad (8)$$

The second derivatives can be replaced by the symmetric Hessian matrix  $\mathbf{H}$  when moving from one dimension to higher number of dimensions. The components of  $\mathbf{H}$  are given by

$$H_{ij} = \frac{\partial^2 u}{\partial x_i \partial x_j}, \quad i, j \in [1, \dots, d], \quad d = \text{spatial dimension} \quad (9)$$

The Hessian matrix can be diagonalized as

$$\mathbf{H} = \mathbf{R} \mathbf{\Lambda} \mathbf{R}^T \quad (10)$$

where  $\mathbf{R}$  is the matrix of the eigenvectors of  $\mathbf{H}$  and  $\mathbf{\Lambda}$  is the diagonal matrix of the eigenvalues of  $\mathbf{H}$ . To have a symmetric positive-definite matrix,  $\mathbf{H}$  can be modified into the metric  $\mathbf{M}$  by taking the absolute value of its eigenvalues such that

$$\mathbf{M} = \mathbf{R} |\mathbf{\Lambda}| \mathbf{R}^T = \mathbf{S} \mathbf{S}^T \quad (11)$$

where  $\mathbf{S} = \mathbf{R} \sqrt{|\mathbf{\Lambda}|}$ . Here,  $\mathbf{M}$  is a function of the co-ordinates  $x$ , so that we have a Riemannian space. Hence using a transformation  $\mathbf{S}$ , a directionally stretched mesh from the original uniform mesh can be obtained. However, a mesh with edges of equal length is desired in the transformed plane  $\mathbf{S}^T$ , where the length of an edge  $\gamma$  is given by

$$e(\gamma) = \int_0^1 \sqrt{\mathbf{s}'(l)^T \mathbf{M}(l) \mathbf{s}'(l)} dl \quad (12)$$

Here,  $\mathbf{s}'(l)$ ,  $l \in [0, 1]$  is the parametric representation of  $\gamma$ . Equation 12 defines a Riemannian metric, where the length of an edge  $\gamma$  is the measure of the interpolation error over the edge. Thus

$$e(x_i - x_j) = \int_0^1 \sqrt{(x_i - x_j)^T \mathbf{M}(x_i - x_j)} dl \quad (13)$$

where  $x_i$  and  $x_j$  are the coordinates of the edge nodes.

The modified Hessian matrix  $\mathbf{M}$  is stored at the background mesh after computation and during the adaptive cycle, its value at any position of the domain can be interpolated on this background mesh.

The error estimation is followed by the adaptive strategy which is based on the spring analogy where the mesh can be viewed as a network of springs whose stiffness constants describe the edge based error  $e$  as shown in the Figure 2. As shown in Ait-Ali-Yahia et al (Ait-Ali-Yahia et al. 1996), the solution of an energy minimization problem can determine the position of the grid vertices. Thus for the node  $x_j$



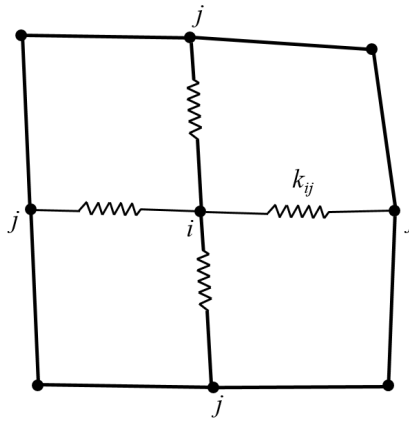


Figure 2. Spring analogy for the mesh movement.

$$\min_{x_j} P(x_j) = \min_{x_j} \sum_{i=1}^n (x_i - x_j)^2 k_i(x_j) \quad (14)$$

where  $k_i(x_j)$  is the spring constant between  $x_i$  and  $x_j$ . These constants can be written as

$$k_i(x_j) = \frac{e(x_i - x_j)}{\|(x_i - x_j)\|} \quad (15)$$

where  $e(x_i - x_j)$  is the length of the edge (Equation 13) defined in the Riemannian field and  $\|(x_i - x_j)\|$  is its Euclidean length. Equation 14 can be simplified to get a system describing the equilibrium state of spring network. This system of equations is solved and the position of the node  $x_j$  is updated by the following relation

$$x_j^{n+1} = x_j^n + \omega \frac{\sum_{i=1}^n (x_i - x_j^n) k_i(x_j^n)}{\sum_{i=1}^n k_i(x_j^n)} \quad (16)$$

where  $\omega$  is the relation factor. The flow diagram of the adaptive procedure is shown in the Figure 3. The grid adaptation and the flow solver are placed in an iterative loop which stops only when a user specified error tolerance is reached.

The block boundary nodes are flagged and their connectivity information is saved before adaptation. After the adaptation, the new position of the block boundary nodes is written out and the mesh is regenerated.

#### 4. Results

In this section, various automatic blocking methods are compared using two different geometries. These blocking techniques are based on medial axis and Cartesian fitting methods. In addition to the automated blocking methods, the results from the manually created block topologies are also included for comparison. These hand-crafted block topologies were independently generated prior to the start of this current work. The adjoint error estimation procedure is employed for the assessment. The objective functions chosen for all the cases is the total pressure loss (as percentage

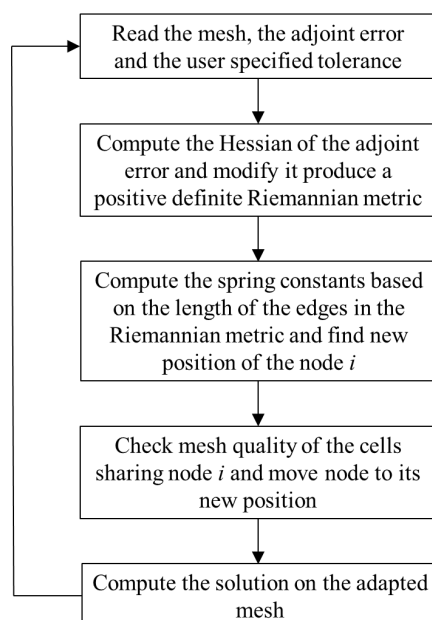


Figure 3. Mesh adaptation flow diagram.

of total inlet pressure). The error is computed with respect to a fine mesh which is a one-step uniform refinement of the original coarse mesh.

The response of the different numerical schemes to the mesh quality might not be the same and a varied degree of tolerance to the mesh quality could be observed. Hence, one of the aims in this section is to find out whether the grading of the block topologies remains consistent if a different numerical scheme is used. Therefore two numerical schemes (Roe and AUSM) have been used here. The cases used in this section include a  $90^\circ$  sector engine intake rig and a rim seal with a blade geometry. The error analysis is described in the following sections.

#### 4.1. Engine intake rig

The  $90^\circ$  sector engine intake rig geometry is shown in the Figure 4. This experimental setup has been used to study the intake lip flow in cross winds (Oriji 2014; Judge and Hynes 2011).

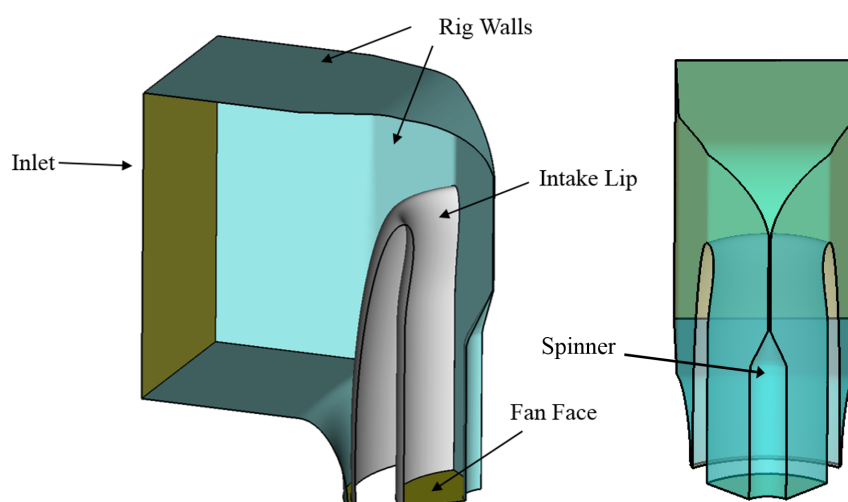


Figure 4. The  $90^\circ$  sector intake rig geometry.

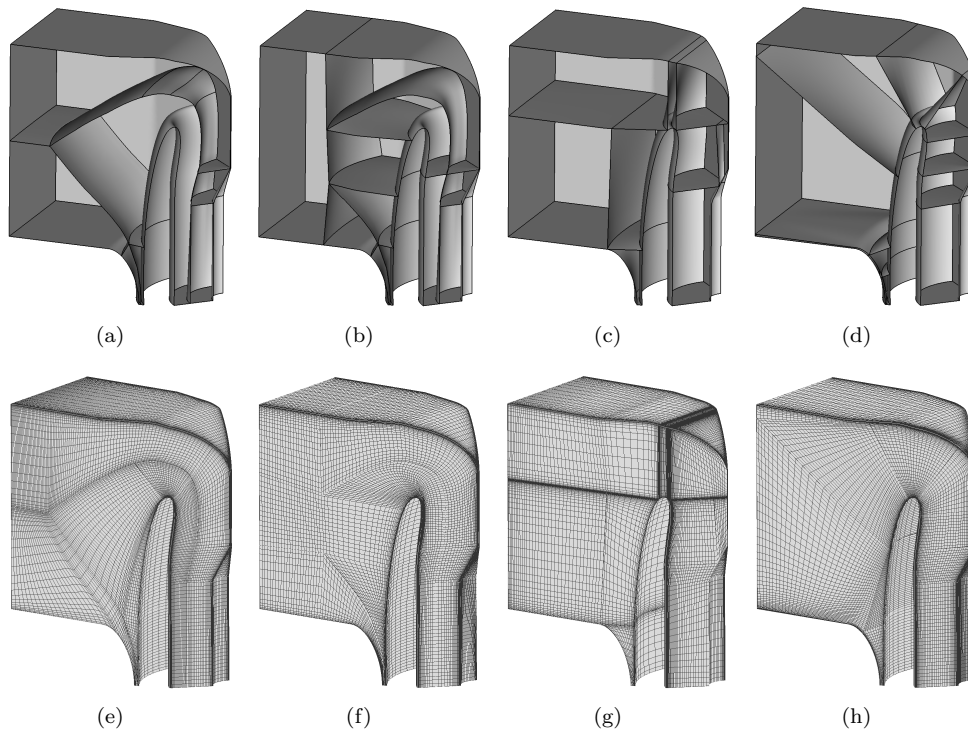


Figure 5. 3D engine intake rig: (a) d-MAT blocking (b) Topmaker blocking (c) Cartesian fitting blocking (d) Manual blocking (e) d-MAT mesh (f) Topmaker mesh (g) Cartesian fitting mesh (h) Manual mesh.

The domain is partitioned using four different block topologies. For the 3D medial axis generation, the Voronoi diagram based algorithm of Dey and Zhao (Dey and Zhao 2004) is used. The d-MAT and TopMaker block topologies are then completed using additional rules (see Appendix). The Cartesian fitting blocking is manually followed. The hand crafted blocking used by Oriji (Oriji 2014) has also been included here for comparison. The block topologies and the resulting meshes are shown in the Figures 5 where a substantial difference in the blocking layout can be observed. The d-MAT approach produces blocks resulting in a mesh which is aligned with the flow direction, especially around the intake lip. The TopMaker blocking also tries to align with the flow direction. The Cartesian fitting method aligns the block topology in the Cartesian frame. Also, from this blocking, the effect of cell clustering to resolve the boundary layers extends out into the main flow path. The coarse mesh cell count is given in the Table 2.

A range of simulations using this setup was carried out in a recent study with varying exit Mach number ( $Ma_{ex}$ ) (Oriji 2014). The pre-separated and post-separated intake flows were investigated including the effect of surface roughness. Here a pre-separation case ( $Ma_{ex} = 0.42$ ) is used. The operating conditions are given in the Table 1. The Reynolds number based on the exit diameter ( $Re_D$ ) is approximately  $7 \times 10^5$ . The simulations are performed using the Spalart-Allmaras (SA) turbulence model, with near wall grid spacing giving  $y^+ \approx 1$ . The axial velocity contours around the intake lip, using the Roe's scheme, are shown in the Figure 6(a) along with the flow streamlines. The adjoint flow is then computed with the total pressure loss as the objective function. The contours of the adjoint Spalart variable are displayed in the Figure 6(b) where a high sensitivity along the intake lip can be seen.

The adjoint error estimate results for both numerical schemes are given in the Table 2. The error in computing the objective function is estimated using the method described above. The overall error can then be obtained by summing individual error contributions. If  $E_j$  is the estimated error,

Table 1. 3D engine intake rig operating conditions.

Exit Mach number	0.42
Inlet stagnation pressure	235073 Pa
Inlet static pressure	234763 Pa
Inlet temperature	314.07 K
Exit pressure	207996 Pa

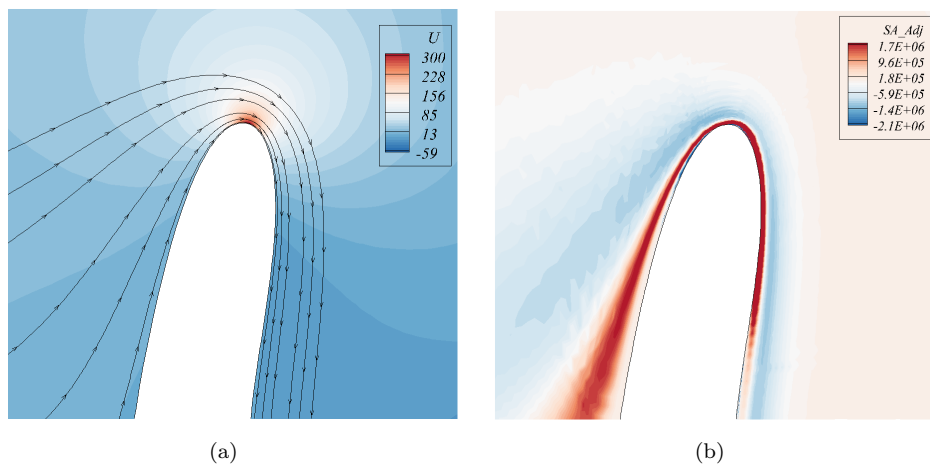


Figure 6. 3D engine intake rig: contours of (a) Axial velocity (b) adjoint counterpart of the Spalart variable around the intake lip.

then the total weighted error  $TE$  over the whole domain for  $N$  number of cells is given by

$$TE = \frac{\sum_{j=0}^N (|E_j| \times Vol_j)}{\sum_{j=0}^N Vol_j} \quad (17)$$

where  $Vol$  is the cell volume (area in 2D). A comparison of the error estimates can then be made by normalizing the total error with the d-MAT error  $TE_{dMAT}$  such that the normalized value  $TE_N$  is given by

$$TE_N = \frac{TE}{TE_{dMAT}} \quad (18)$$

The results indicate that the medial axis based meshes provide the most accurate value of the objective function than the other block topologies. One of the reasons for this is that for the similar cell count, the medial axis based approach generates a mesh which has better alignment with the flow and also a more uniform cell size distribution in the main flow path than the other approaches. The Cartesian fitting mesh is clustered in the main flow path. This constraint effects the overall uniformity of the mesh size producing coarser mesh in the areas away from the block boundaries, thus contributing to the error. The trend is consistent for both numerical schemes. It is also found that the AUSM scheme gives a slightly less error (approximately 10%) for all block topologies when compared with their counterparts in the Roe scheme.

The adjoint error maps for all the meshes and both numerical schemes are shown in the Figure 7. These are taken at  $z=0$  slice where a consistent error pattern for both the schemes can be seen. The cell skewness histograms for various meshes are displayed in the Figure 8. When looking at the mesh quality from the viewpoint of the geometrical mesh quality metrics such as skewness, the d-MAT mesh performs badly despite providing the more accurate solution. This shows a lack of connection between the solution accuracy and the traditional quality metrics implying that the quality is mainly dependent on the physical solution. This has also been demonstrated in

Table 2. 3D engine intake rig: coarse mesh cell count and the total normalized error.

Blocking Type	$N$ (Approx)	$TE_N$ (Roe)	$TE_N$ (AUSM)
d-MAT	450,000	1.0	1.0
TopMaker	430,000	1.01	0.9
Cart.Fitting	450,000	1.5	1.5
Manual blocking	440,000	1.3	1.2

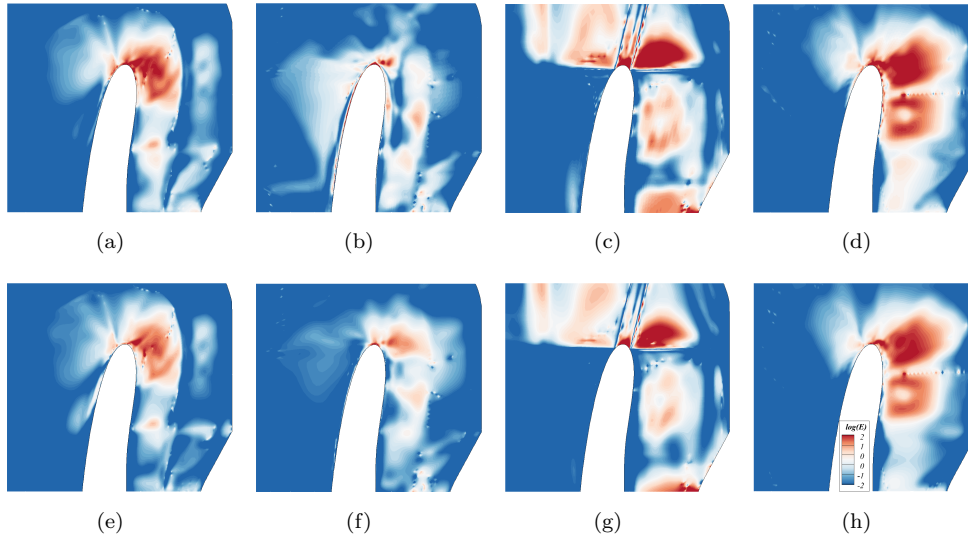


Figure 7. The error contours near the intake lip ((a)-(d) Roe scheme, (e)-(h) AUSM scheme): (a and e) d-MAT (b and f) TopMaker (c and g) Cartesian fitting (d and h) manual.

the literature (Knupp 2007). Nevertheless, the mesh quality can have a strong influence on the solution convergence and can be used to improve defective meshes. The results are also compared with the experimental data as shown in the Figure 9. Here the  $Ma_{is}$  along the lip surface center line is plotted against the dimensionless lip length,  $L_D$ . This length is non-dimensionalized by the distance from the lip highlight to the fan face. A better agreement with the measurements can be seen for the medial axis meshes. Similar error analysis was performed in (Ali 2015) for more smooth meshes of a two-dimensional (2D) labyrinth seal case. It was demonstrated that even for elliptically smoothed 2D meshes, the medial axis based approaches perform better than the other techniques.

#### 4.2. Rim seal cavity with blade

The second case used in this study is a rim seal cavity attached to a turbine blade as shown in the Figure 10. Flow from the cavity is injected into the main annulus to control the leakage of flow from the hub gap between the stator and rotor disks. The blade and main gas flow path geometry of Langston's experiment (Langston, Nice, and Hooper 1977) is used here.

The rim seal block topologies are shown in Figure 11. The conventional H-O-H block topology is used for meshing the main gas flow path as shown in the Figure 12. Figure 13(a) shows the main flow axial velocity contours whereas adjoint axial velocity contours for both numerical schemes are shown in the Figure 13(b) and 13(c). The adjoint sensitivity of the AUSM scheme is slightly less than that of the Roe scheme. A Similar trend can be seen in the Figures 14(b) and 14(c) where the SA adjoint sensitivity contours using both schemes are displayed. Figure 14(a) shows the primal flow in the rim seal.

The results of the adjoint error estimation along with the mesh sizes for different block topologies are shown in the Table 3. As depicted by the statistics, the manually generated mesh produces more

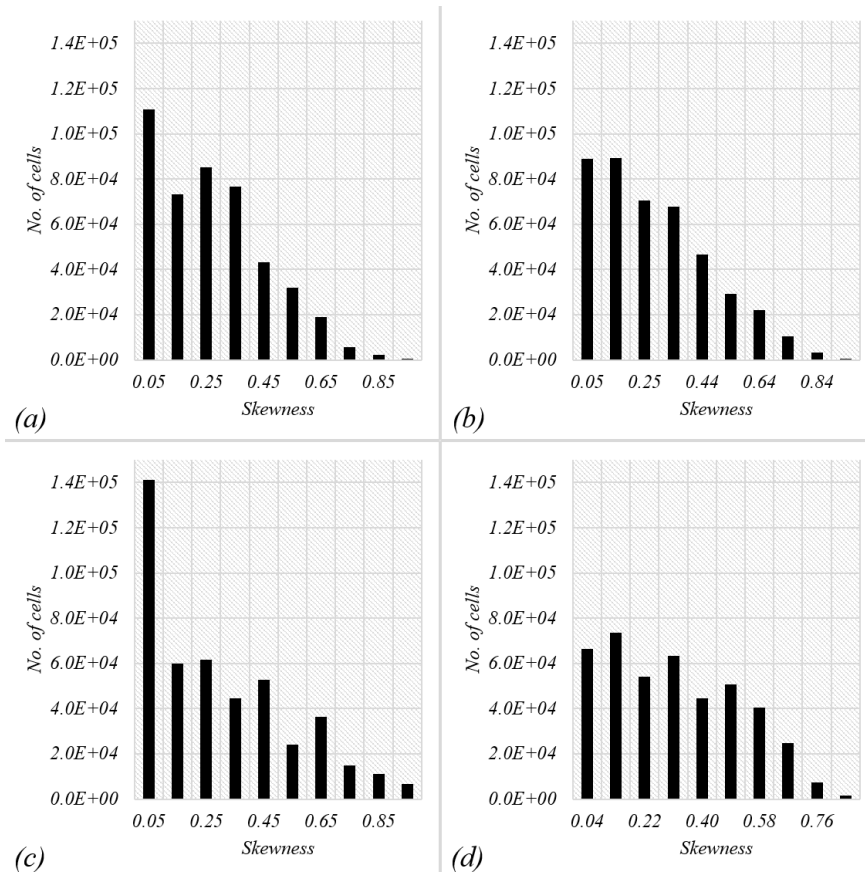


Figure 8. 3D engine intake: Skewness histogram (a) d-MAT (b) TopMaker (c) Cartesian fitting (d) manual.

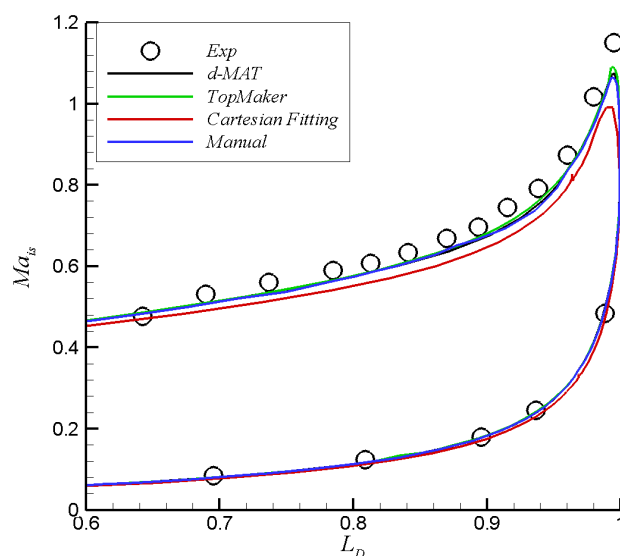


Figure 9. Lip isentropic Mach number profiles.

accurate functional estimates than the rest of the methods. This is followed by the d-MAT mesh. The results are similar for both numerical schemes. Figure 15 show the mesh alignment with the flow for various approaches. The manually generated mesh performs well here due it smoothness and better flow alignment. The d-MAT and the TopMaker meshes are less aligned with the flow as compared to the manual mesh. Less degree of mesh smoothness can also be observed in these



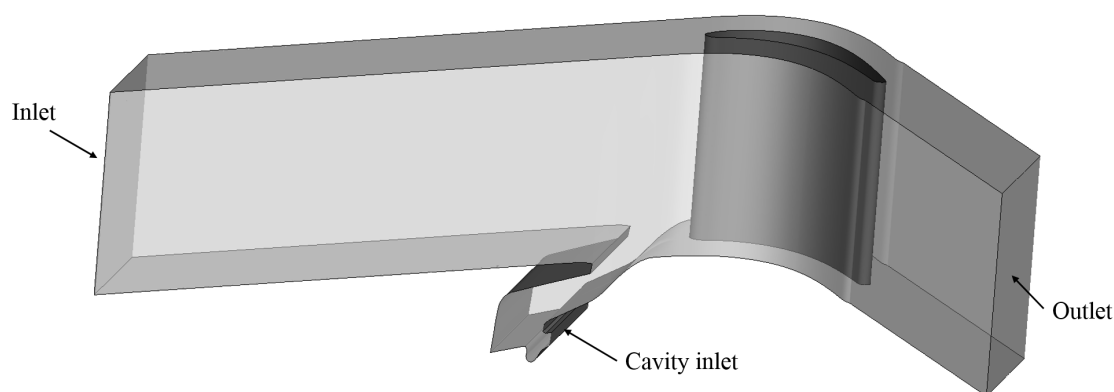


Figure 10. Rim seal cavity with blade.

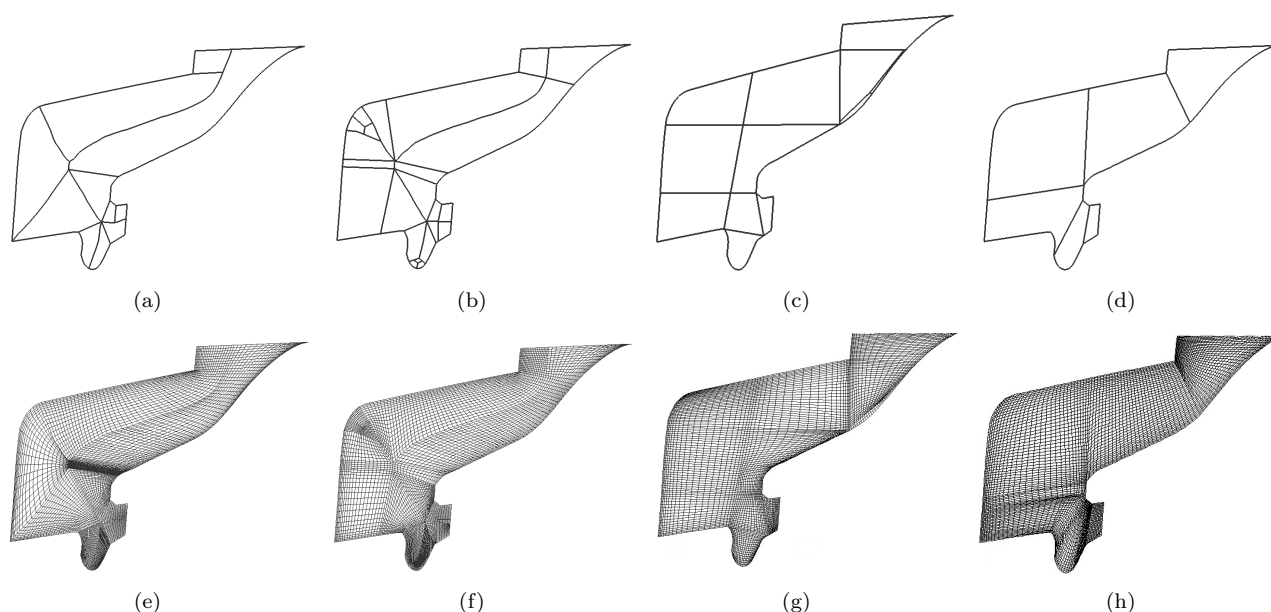


Figure 11. Rim seal: (a) d-MAT blocking (b) Topmaker blocking (c) Cartesian fitting blocking (d) Manual blocking (e) d-MAT mesh (f) Topmaker mesh (g) Cartesian fitting mesh (h) manual mesh.

meshes. The Cartesian meshes also struggles to achieve uniform cell size distribution and the alignment with the flow thus resulting in the highest error.

The adjoint error maps of the blade and the rim seal for all the block topologies are shown in Figure 16 and 17 respectively. Less error (around 10 %) using the AUSM scheme is observed on all the block topologies when compared to the Roe scheme, as displayed in the error maps. The pressure coefficient plot at the 50% span is shown in the Figure 18 and compared with the experimental data (Langston, Nice, and Hooper 1977). All the block topologies yielded the same profile shown in Figure 18. This is because the same main gas flow path blocking is used for all the cases and the varying block topology in the rim seal does not seem to affect the conditions at the mid span.

The results from cases used in this section shows the superiority of the medial axis block topologies over the other automatic blocking methods even when a different numerical scheme is used. The performance of the AUSM scheme was slightly better than the Roe scheme for the two cases analyzed. However, the comparison between the two schemes cannot lead to a general conclusion, because the flows in these cases did not exhibit features like shocks and shear layers etc.

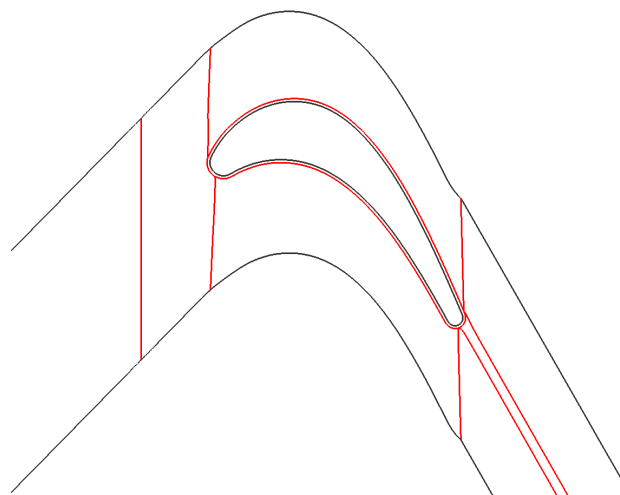


Figure 12. Main gas flow path H-O-H block topology.

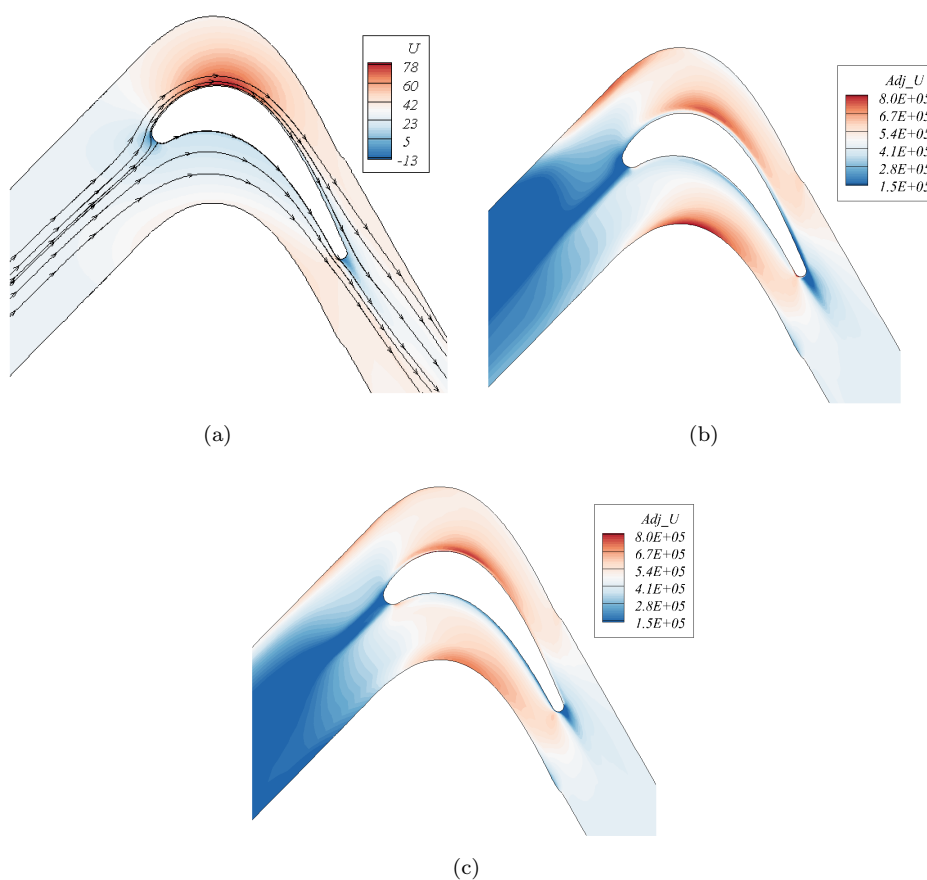


Figure 13. Main gas flow path: Mid span contours of (a) axial velocity (b) adjoint axial velocity for Roe scheme (c) adjoint axial velocity for AUSM scheme.

## 5. Block topology adaptation

The adjoint based error estimation has been employed in the previous section for block topology comparison. In this section, its use to adapt the block topology is presented. The aim is to move the block boundaries through node movement. As stated in the Section 3, this is achieved in an



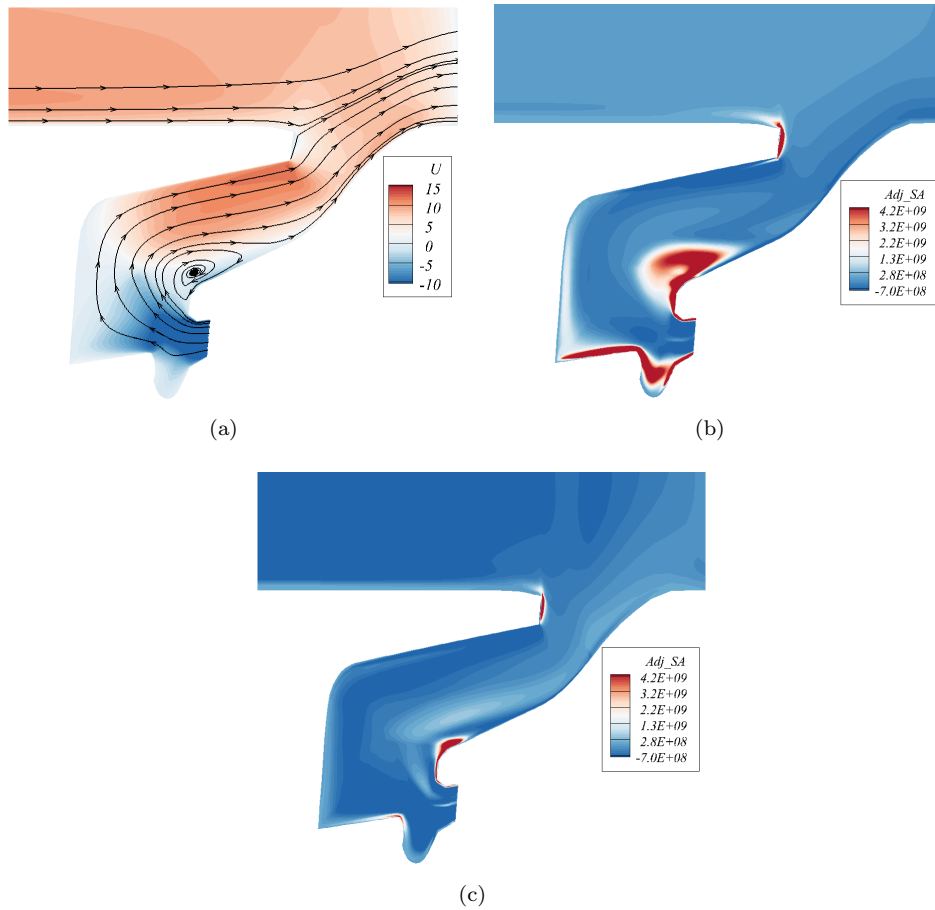


Figure 14. Rim seal: contours of (a) axial velocity (b) adjoint SA for Roe scheme (c) adjoint SA for AUSM scheme.

Table 3. Rim seal with blade: coarse mesh cell count and the total normalized error.

Blocking Type	$N$ (Approx)	$TE_N$ (Roe)	$TE_N$ (AUSM)
d-MAT	540,000	1.0	1.0
TopMaker	540,000	1.01	1.02
Cart.Fitting	560,000	1.3	1.4
Manual blocking	500,000	0.9	0.89

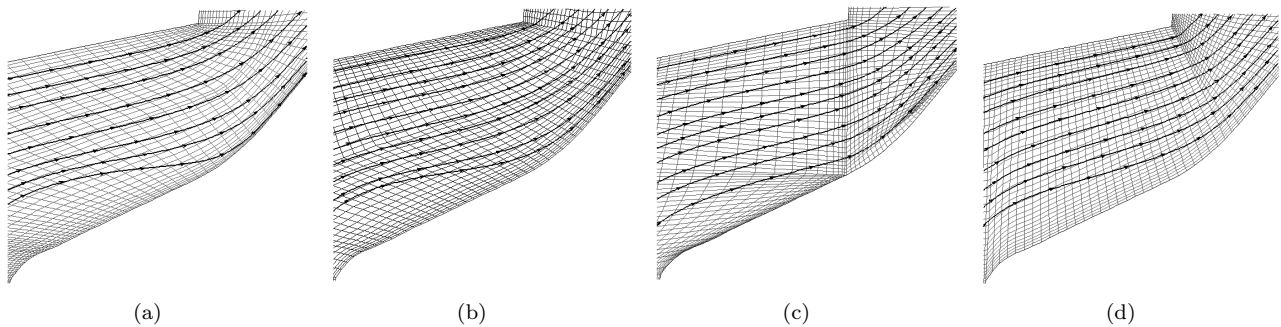


Figure 15. Rim seal: streamlines against the mesh (a) d-mat (b) Topmaker (c) Cartesian fitting (d) Manual.

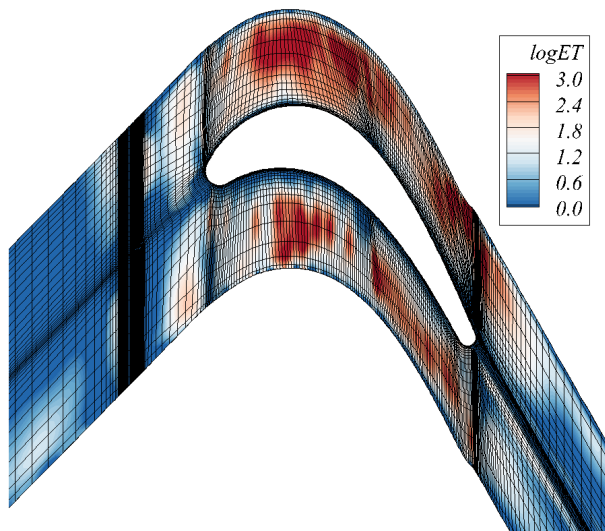


Figure 16. Adjoint error map around the blade.

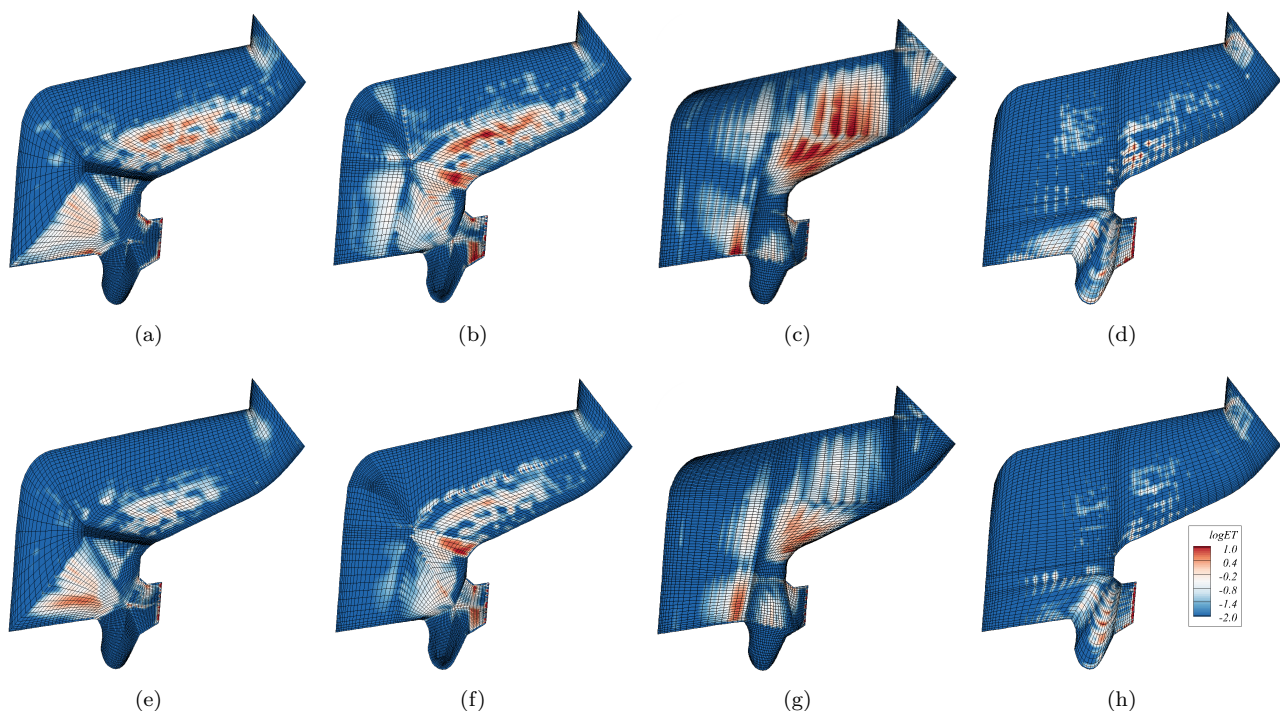


Figure 17. Rim seal error maps ((a)-(d) Roe scheme, (e)-(h) AUSM scheme ): (a and e) d-MAT (b and f) TopMaker (c and g) Cartesian fitting (d and h) manual.

‘implicit’ way which means that whole mesh built on a specific blocking is first adapted through adjoint error based nodal movement and then the nodes representing the block boundaries are extracted and the mesh is regenerated using this adapted block topology.

Traditionally, for flow feature based mesh movement, the Hessian of a selected solution variable (for example Mach number) is used as a measure of error. The adaptation scheme is based on a spring analogy (see section 3) where the mesh nodes are moved until the equilibrium state of the spring forces is reached thus equidistributing the error. In the present implementation, the Hessian of the adjoint error is used for the mesh movement. Hence, the nodes are moved towards the regions having strong adjoint error gradients. The adaptation mechanics is described in the section 3. The rationale behind not using the adapted mesh itself instead of the block topology is

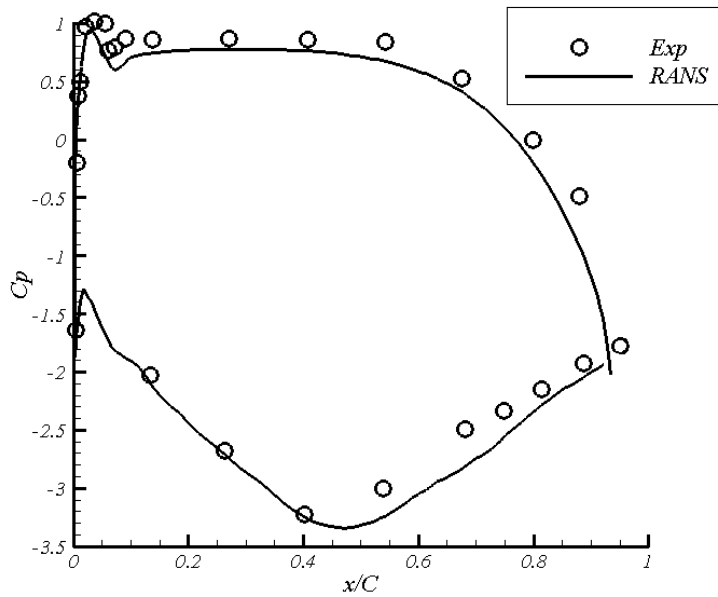


Figure 18. Pressure coefficient profile at the blade mid span.

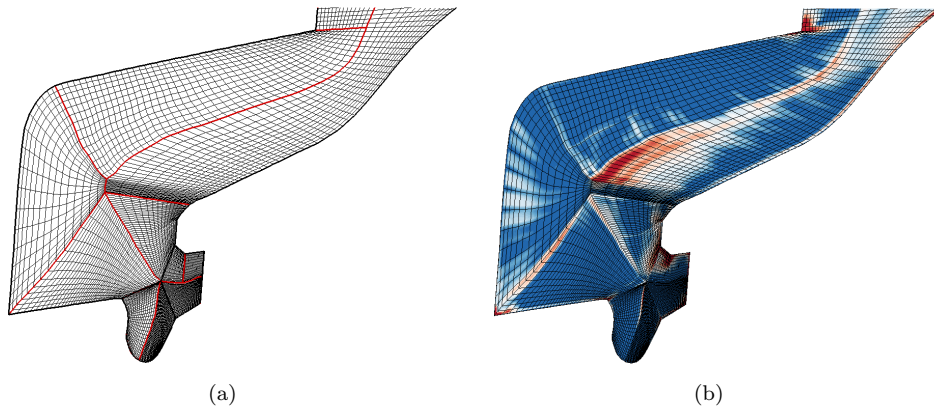


Figure 19. Rim seal: (a) d-MAT original blocking and mesh (b) the adjoint error map.

that the resulting mesh can have low quality cells. Such type of skewed or highly stretched meshes can be problematic for high fidelity simulations like hybrid RANS-LES or LES. In this section, the adaptation technique is demonstrated with the help of two seal geometries where the medial axis based block topologies are adapted.

### 5.1. Rim seal

The first case demonstrated here is the rim seal domain used in the previous section. The block topology and the mesh for the rim seal case are shown in the Figure 19(a). The adjoint error estimation is then carried out on this mesh following the same procedure used in the previous section. The objective function is the total pressure loss. The resulting error map is shown in the Figure 19(b) which highlights the region in the middle of the flow path for modification.

The next step is to perform the block boundary adaptation. The adaptation algorithm strives to move the mesh towards the areas having high gradients of the adjoint error. The adapted block topology represented by the dotted lines is shown in the Figure 20 along with the original blocking

represented by the solid lines.

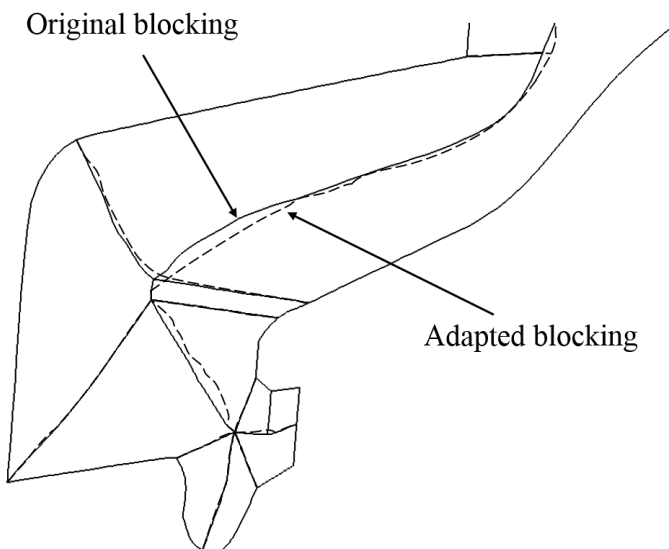


Figure 20. Rim seal: d-MAT original and adapted block topologies.

A mesh of same size is then generated based on the adapted block topology and the adjoint error is recomputed. The mesh based on the adapted topology and the resulting error map are shown in the Figure 21. A decrease of 10% in the total adjoint error resulted as a result of the adaptation.

## 5.2. Labyrinth seal

The labyrinth seal case is now presented. The block topology and the mesh for this domain are shown in the Figure 22(a). The adjoint error estimation is then performed using again the total pressure loss as the objective function. Next, the block boundary adaptation is performed. A smoother and more orthogonal adapted mesh is produced (see Figure 22(b)). The adapted and original block topologies are shown in the Figure 22(c). The total estimated error was reduced by approximately 6%.

The cases demonstrated above show that a block topology adaptation scheme can be employed to get optimized blocking. The globally adapted whole mesh can result in highly skewed cells and the situation might not improve even after smoothing. This has an impact on the solution convergence. Hence only the adapted nodes representing the block topology are extracted and

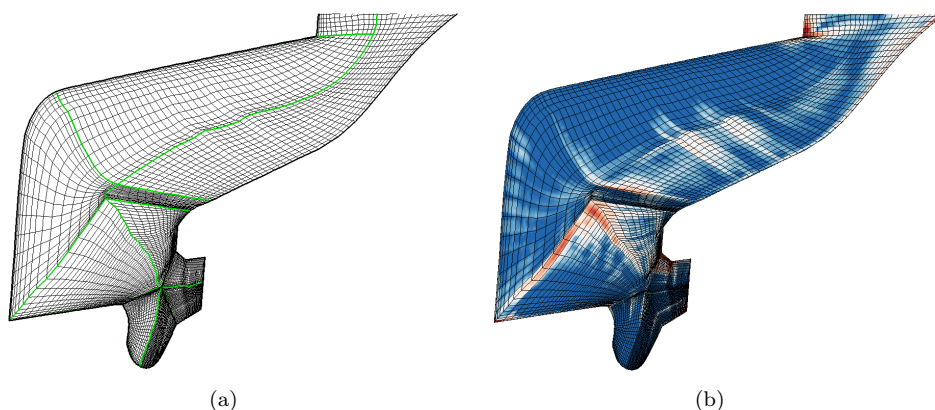


Figure 21. Rim seal: (a) d-MAT adapted blocking and mesh (b) the adjoint error map.



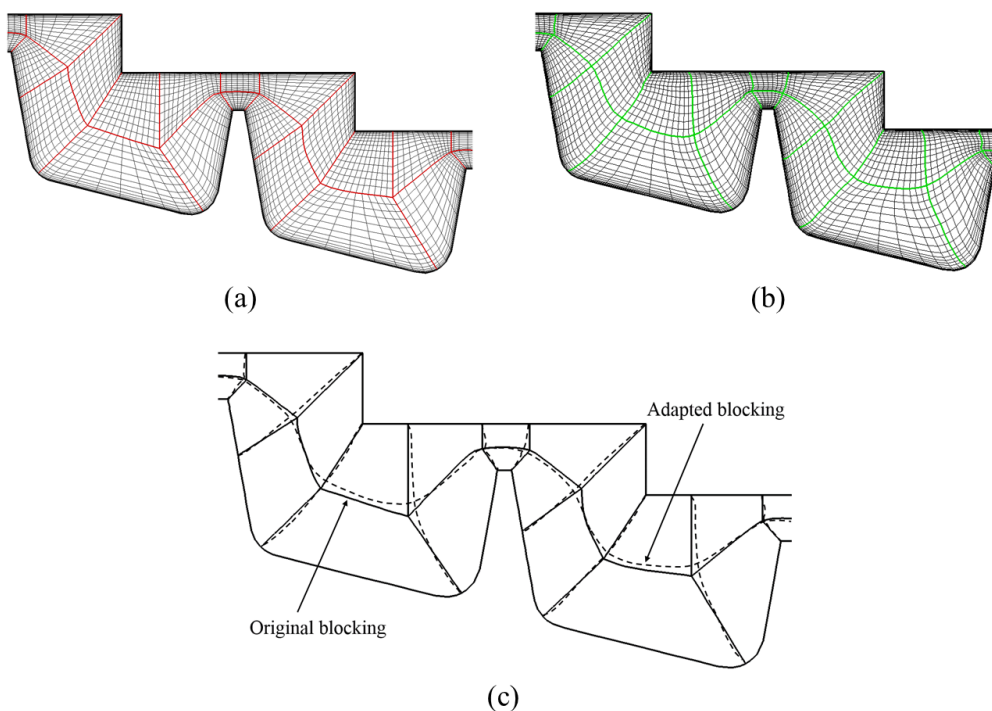


Figure 22. Labyrinth seal: (a) d-MAT original blocking and mesh (b) d-MAT adapted blocking and mesh (c) superimposed original and adapted blockings.

the mesh is regenerated around it. The medial axis block based topology used in these cases (narrow duct type geometries) is already in near optimal form and hence a limited amount of mesh movement could be achieved. Therefore, applying and further developing this technique to a variety of domains can be an area of future research.

## 6. Conclusion

Various automatic and manual blocking methods are compared in the first part of this paper. The adjoint error estimation performed on the meshes generated over various block topologies shows that, in general, medial axis based methods produce optimal domain decomposition. This is due to the fact that a better flow alignment and a more uniform cell size distribution is produced by these techniques. *It should, however, be noted that the initial setup and the experience in implementing these automatic techniques can impact the final outcome.* Medial axis based technique can be useful for many internal flows where the inlet flow angle remains constant. We have also observed that the manually generated block topologies yield better results in some cases while not performing the best in the others. This can also be attributed to the experience of the user. Thus the automated approaches can guide a CFD practitioner to optimally block the domain with the advantage of being less user intensive. The use of the adjoint error estimation for block topology adaptation was also explored in the second part of the paper. Such a strategy can be useful to further improve the block topology created using an existing automatic blocking method (for example, using the d-MAT approach in this paper). The results show a potential in such an adaptation scheme to be used and developed further.

## Acknowledgments

[deleted to maintain the integrity of the review process].

## Funding

[deleted to maintain the integrity of the review process].

## References

- Ait-Ali-Yahia, D., W. G. Habashi, A. Tam, M-G. Vallet, and M. Fortin. 1996. "A directionally adaptive methodology using an edge-based error estimate on quadrilateral grids." *International journal for numerical methods in fluids* 23 (7): 673–690.
- Ali, Z. 2015. "Optimal block topology generation for CFD meshing." Ph.D. thesis. Department of Engineering, University of Cambridge, UK.
- Baker, T. J. 1997. "Mesh adaptation strategies for problems in fluid dynamics." *Finite Elements in Analysis and Design* 25 (3): 243–273.
- Baskett, L., and R. Haimes. 2001. "Feature extraction of shear layers." In *AIAA Computational Fluid Dynamics Conference, 15 th, Anaheim, CA*, .
- Blacker, T. D., and R. J. Myers. 1993. "Seams and Wedges in Plastering: A 3D Hexahedral Mesh Generation Algorithm." *Engineering with Computers* 2: 83–93.
- Bommes, D., T. Lempfer, and L. Kobbelt. 2011. "Global structure optimization of quadrilateral meshes." In *Computer Graphics Forum*, Vol. 30375–384. Wiley Online Library.
- Danielsson, P-E. 1980. "Euclidean distance mapping." *Computer Graphics and image processing* 14 (3): 227–248.
- Dey, T. K., and W. Zhao. 2004. "Approximate medial axis as a Voronoi subcomplex." *Computer-Aided Design* 36 (2): 195–202.
- Duta, M. C., S. Shahpar, and M. B. Giles. 2007. "Turbomachinery design optimization using automatic differentiated adjoint code." In *ASME Turbo Expo 2007: Power for Land, Sea, and Air*, 1435–1444. American Society of Mechanical Engineers. ASME Paper No. GT 2007-28329.
- Dwigh, R. P. 2008. "Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation." *Journal of Computational Physics* 227 (5): 2845–2863.
- Dwight, R.P. 2007. "Goal-oriented mesh adaptation using a dissipation-based error indicator." In *Proceedings of ICFD Conference*, University of Reading.
- Fidkowski, K.J. 2007. "A Simplex Cut-Cell Adaptive Method for High-Order Discretizations of the Compressible NavierStokes Equations." Ph.D. thesis. Massachusetts Institute of Technology.
- Fidkowski, K.J., and D.L. Darmofal. 2011. "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics." *AIAA Journal* 49(4): 673–694.
- Fogg, H. J., C. G. Armstrong, and T. T. Robinson. 2015a. "Automatic generation of multiblock decompositions of surfaces." *International Journal for Numerical Methods in Engineering* 101 (13): 965–991.
- Fogg, H. J., C. G. Armstrong, and T. T. Robinson. 2015b. "Enhanced medial-axis-based block-structured meshing in 2-D." *Computer-Aided Design* .
- Giles, M. B., M. C. Duta, J-D Müller, and N. A. Pierce. 2003. "Algorithm developments for discrete adjoint methods." *AIAA journal* 41 (2): 198–205.
- Giles, M. B., and N. A. Pierce. 2002a. "Adjoint Error Correction for Integral Outputs." *Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, Springer 25.
- Giles, M. B., and N. A. Pierce. 2002b. "Adjoint Error Correction for Integral Outputs." *Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, Springer 25.
- Habashi, W. G., J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M-G. Vallet. 2000. "Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD. Part I: General principles." *International Journal for Numerical Methods in Fluids* 32 (6): 725–744.
- Huang, J., Y. Tong, H. Wei, and H. Bao. 2011. "Boundary aligned smooth 3D cross-frame field." In *ACM Transactions on Graphics (TOG)*, Vol. 30143. ACM.
- Judge, M., and T. Hynes. 2011. "Experimental Investigation into Effect of Roughness on Crosswind Separation." Report, Whittle Lab, University of Cambridge, UK: 1–81.
- Knupp, P. 2007. "Remarks on mesh quality." In *45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, USA*, 7–10.

- 1 Kowalski, N., F. Ledoux, and P. Frey. 2013. "A PDE based approach to multidomain partitioning and
- 2 quadrilateral meshing." In *Proceedings of the 21st international meshing roundtable*, 137–154. Springer.
- 3 Kowalski, N., F. Ledoux, and P. Frey. 2014. "Block-Structured Hexahedral Meshes for CAD Models using
- 4 3D Frame Fields." *Procedia Engineering* 82: 59–71.
- 5 Langston, L., M. Nice, and R. Hooper. 1977. "Three-dimensional flow within a turbine cascade passage."
- 6 *Journal of Engineering for Gas Turbines and Power* 99 (1): 21–28.
- 7 Lapworth, L. 2004. "Hydra-CFD: a framework for collaborative CFD development." In *International Con-*
- 8 *ference on Scientific and Engineering Computation (IC-SEC), Singapore, June*, Vol. 30.
- 9 Li, Y., Y. Liu, W. Xu, W. Wang, and B. Guo. 2012. "All-hex meshing using singularity-restricted field."
- 10 *ACM Transactions on Graphics (TOG)* 31 (6): 177.
- 11 Li, Y., W. Wang, R. Ling, and C. Tu. 2011. "Shape optimization of quad mesh elements." *Computers &*
- 12 *Graphics* 35 (3): 444–451.
- 13 Liou, M-S., and C. J. Steffen. 1993. "A new flux splitting scheme." *Journal of Computational physics* 107
- 14 (1): 23–39.
- 15 Malcevici, I. 2011. "Automated Blocking for Structured CFD Gridding with an Application to Turboma-
- 16 chinery Secondary Flows." In *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii.
- 17 Milli, A., and S. Shahpar. 2012. "PADRAM: Parametric design and rapid meshing system for complex tur-
- 18 bomachinery configurations." In *ASME Turbo Expo 2012: Turbine Technical Conference and Exposition*,
- 19 2135–2148. American Society of Mechanical Engineers. ASME Paper No. GT 2012-69030.
- 20 Moinier, P. 1999. "Algorithm developments for an unstructured viscous flow solver." Ph.D. thesis. Oxford
- 21 University.
- 22 Muller, J.D., and M.B. Giles. 2001. "Solution Adaptive Mesh Refinement Using Adjoint Error Analysis."
- 23 *AIAA Paper* 2001-2550.
- 24 Nemec, M., and M. J. Aftosmis. 2007. "Error Estimation and Adaptive Refinement for embedded-Boundary
- 25 Cartesian Meshes." *AIAA Paper* 2007-4187.
- 26 Orijji, U. R. 2014. "Numerical investigation of intake flows in crosswinds." Ph.D. thesis. Department of
- 27 Engineering, University of Cambridge.
- 28 Owen, M. L. Staten S. J., and T. D. Blacker. 2005. "Unconstrained paving and plastering: A new idea for all
- 29 hexahedral mesh generation." In *Proceedings of 14th International Meshing Roundtable*, 399416. Sandia
- 30 National Lab.
- 31 Palacios, J., and E. Zhang. 2007. "Rotational symmetry field design on surfaces." In *ACM Transactions on*
- 32 *Graphics (TOG)*, Vol. 2655. ACM.
- 33 Park, M.A. 2004. "Adjoint-Based, Three Dimensional Error Prediction and Grid Adaptation." *AIAA Journal*
- 34 42: 18541862.
- 35 Pirzadeh, S.Z. 1999. "An adaptive unstructured grid method by grid subdivision, local remeshing, and grid
- 36 movement." *AIAA paper* AIAA Paper No. 99-3255.
- 37 Price, M.A., and C.G. Armstrong. 1995a. "Hexahedral Mesh Generation by Medial Surface Subdivision: Part
- 38 I. Solids with Convex Edges." *International Journal for Numerical Methods in Engineering* 38: 33353359.
- 39 Price, M.A., and C.G. Armstrong. 1995b. "Hexahedral Mesh Generation by Medial Surface Subdivision: Part
- 40 II. Solids with Flat and Concave Edges." *International Journal for Numerical Methods in Engineering*
- 41 38: 33353359.
- 42 Quadros, W. R. 2014. "LayTracks3D: a new approach to meshing general solids using medial axis transform."
- 43 *Procedia Engineering* 82: 72–87.
- 44 Ragnemalm, I. 1993. "The Euclidean distance transform in arbitrary dimensions." *Pattern Recognition*
- 45 *Letters* 14 (11): 883–888.
- 46 Rigby, D.L. 2004. "A technique for automatic Multi-Block topology generation using the medial axis."
- 47 *NASA/CR FEDSM2003-45527*.
- 48 Roy, C.J. 2009. "Strategies for Driving Mesh Adaptation in CFD." *AIAA Paper* 2009-1302.
- 49 Roy, C. J. 2010. "Review of Discretization Error Estimators in Scientific Computing." *AIAA* 2010-126.
- 50 Shahpar, S., and L. Lapworth. 2003. "PADRAM: Parametric design and rapid meshing system for turbo-
- 51 machinery optimisation." In *ASME Turbo Expo 2003, collocated with the 2003 International Joint Power*
- 52 *Generation Conference*, 579–590. American Society of Mechanical Engineers. ASME Paper No. GT 2003-
- 53 38698.
- 54 Sheehy, D.J., C.G. Armstrong, and D.J. Robinson. 1995. "Computing the medial surface of a solid from a
- 55 domain Delauney triangulation." In *ACM Symposium on Solid Modeling Foundations and Applications*,
- 56 201–212.
- 57
- 58
- 59
- 60

- Tam, T.K.H., and C.G. Armstrong. 1991. "2D Finite Element Mesh Generation by Medial Axis Subdivision." *Adv. Eng. Software* 13: 313324.
- Tucker, P.G. 2003. "Differential equation-based wall distance computation for DES and RANS." *Journal of Computational Physics* 190(1): 229248.
- Venditti, D.A., and D.L. Darmofal. 2002a. "Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flow." *Journal of Computational Physics* 176: 40–69.
- Venditti, D. A., and D. L. Darmofal. 2002b. "Grid adaptation for functional outputs: Application to two-dimensional inviscid flows." *Journal of Computational Physics* 176 (1): 40–69.
- Vleugels, J., and M. Overmars. 1995. "Approximating generalized Voronoi diagrams in any dimension." Technical report UU-CS-1995-14 Utrecht University.
- Warren, G. P., W. K. Anderson, J. L. Thomas, and S. L. Krist. 1991. "Grid convergence for adaptive methods." *AIAA paper* AIAA Paper No. 91-1592.
- White, F. M., and I. Corfield. 2006. *Viscous fluid flow*. Vol. 3. McGraw-Hill New York.
- Xia, H., and P.G. Tucker. 2010. "Finite volume distance field and its application to medial axis transforms." *International Journal for Numerical Methods in Engineering* 82(1): 114134.
- Xia, H., and P.G. Tucker. 2011. "Fast equal and biased distance fields for medial axis transform with meshing in mind." *Applied Mathematical Modelling* 35: 580419.

## Appendix

### Error Estimation

Consider a computational domain  $\Omega$  and let  $\Omega_H$  and  $\Omega_h$  are the coarse mesh and the fine mesh discretizations of this domain respectively.  $H$  and  $h$  ( $H > h$ ) here represent the mesh length scales associated with a particular discretization such as finite difference or finite volume. The coarse mesh has a mesh density that is able to capture the basic feature of the flow but might not yield the desired level of accuracy. The fine mesh on the other hand is a systematic uniform refinement of the coarse mesh which can provide the accuracy required but is computationally expensive. The discretization of the governing equations on the coarse and fine mesh yield the residual vectors which can be denoted by  $R_H(Q_H)$  and  $R_h(Q_h)$  respectively where  $Q$  is the solution of system of the governing flow equations. Let  $J(Q)$  be the objective function which one wants to estimate. The estimation of this objective function on the coarse and fine mesh are denoted by  $J_H(Q_H)$  and  $J_h(Q_h)$ . An estimate of  $J_h(Q_h)$  on the fine grid, without solving on the fine grid, can be made by a Taylor's series expansion of  $J_h(Q_h)$  about the solution on the coarse grid

$$J_h(Q_h) = J_h(Q_h^H) + \left. \frac{\partial J_h}{\partial Q_h} \right|_{Q_h^H} (Q_h - Q_h^H) + \dots \quad (19)$$

Also the expansion of  $R_h(Q_h)$  about the coarse mesh yields

$$R_h(Q_h) = R_h(Q_h^H) + \left. \frac{\partial R_h}{\partial Q_h} \right|_{Q_h^H} (Q_h - Q_h^H) + \dots \quad (20)$$

The vector  $(Q_h^H)$  in the above equation is the coarse mesh solution estimated on the fine mesh with a suitably defined prolongation operator. The vector  $\left. \frac{\partial J_h}{\partial Q_h} \right|_{Q_h^H}$  in Eqn. (19) represents the linear sensitivity of the fine mesh function with respect to  $(Q_h^H)$ . Moreover, the vector  $\left. \frac{\partial R_h}{\partial Q_h} \right|_{Q_h^H}$  in Eqn. (20) is the fine mesh Jacobian evaluated using the projected coarse mesh solution.  $(Q_h^H)$  can be evaluated by

$$Q_h^H = I_h^H Q_H \quad (21)$$



where  $I_h^H$  represents an appropriate prolongation operator. For example the coarse grid solution can be reconstructed on the fine mesh using linear or higher order interpolation. Assuming the well-posedness, the Eqn. (20) can be inverted. Also knowing that  $R_h(Q_h) = 0$  for steady state problem, Eqn. (20) yields

$$(Q_h - Q_h^H) \approx - \left\{ \frac{\partial R_h}{\partial Q_h} \Big|_{Q_h^H} \right\}^{-1} R_h(Q_h^H) \quad (22)$$

From Equations (22) and (19) we get

$$J_h(Q_h) \approx J_h(Q_h^H) - (\nu_h|_{Q_h^H})^T R_h(Q_h^H) \quad (23)$$

where  $\nu_h|_{Q_h^H}$  is the discrete adjoint solution vector estimated at the fine mesh using  $Q_h^H$ . This adjoint solution vector satisfies the equation

$$\left\{ \frac{\partial R_h}{\partial Q_h} \Big|_{Q_h^H} \right\}^T \nu_h|_{Q_h^H} = \left\{ \frac{\partial J_h}{\partial Q_h} \Big|_{Q_h^H} \right\}^T \quad (24)$$

Eqn. (23) requires the evaluation of the term  $(\nu_h|_{Q_h^H})^T$  on the fine grid. To avoid this evaluation,  $(\nu_h|_{Q_h^H})^T$  can also be estimated thorough coarse grid adjoint interpolation onto the fine grid via some projection operator

$$\nu_h^H = I_h^H \nu_H \quad (25)$$

where the coarse mesh adjoint solution  $\nu_H$  is the solution of following adjoint equations on the coarse mesh

$$\left\{ \frac{\partial R_H}{\partial Q_H} \right\}^T \nu_H = \left\{ \frac{\partial J_H}{\partial Q_H} \right\}^T \quad (26)$$

The final estimate of the objective function  $J(Q)$  is then given by

$$J(Q) = J_h(Q_h^H) - (\nu_h^H)^T R_h(Q_h^H) \quad (27)$$

The second expression on the right hand side of the above equation is called the error correction term. It can be seen that the error in the objective function is related to the local residual error through the adjoint variables which act as weight function. Eqn. (23) can be disintegrated into the following form

$$J_h(Q_h^H) - J_h(Q_h) \approx (\nu_h^H)^T R_h(Q_h^H) + (\nu_h|_{Q_h^H} - \nu_h^H)^T R_h(Q_h^H) \quad (28)$$

The first term on the right hand side of Eqn. (28) is the main computable error estimate while the second term is the error in this computable error estimate. This first term on the right hand side shows that the adjoint variables directly relates the error in the given functional to the local residual errors. Stating in another way, the adjoint solution act as a weight function to the local residual error and gives the effect of the residual error on the output functional.

### *Rules for the d-MAT Blocking*

The rules for d-MAT blocking are noted below. Some of these are borrowed from the TopMaker method.

**Rule 1 (R1)** Expansion features are connected to the nearest 'medial vertex'; that is if that point lies within sight of the fan caused by the expansion feature. If not, it is connected to the nearest point on the shock feature topology.

**Rule 2 (R2)** Hanging features (features which are not connected to anything) are extended to the nearest point on the geometry. The medial axis based blocking of a simple domain using these rules is shown in the Figure A.1

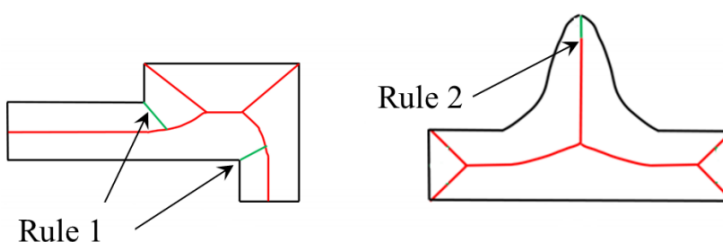


Figure A.1. Rules for d-MAT blocking.